Parallel Computations of the Two-way Wave Equation
雙向波動方程式之平行運算

Shyan-Lung Lin　　Shang-June Su
林賢龍　　蘇賢鈞

逢甲大學自動控制工程學系
Department of Automatic Control Engineering
Feng Chia University

摘要

本研究將在多處理器的環境下，分別使用個別記憶體平行電腦—transputer，與共同記憶體平行電腦—Sequent Symmetry，來求解雙向波動方程式。除運用 Leap Frog、MacCormack、Runge Kutta 等數值方法來進行求解外，並考慮脈波在三種不同的不連續聲速介質下的傳遞情形，以了解其在精確度與強健性的表現。本研究著重於數值方法與運算環境間的關係比較，其中包括雙向波動方程式在不同平行處理環境下的平行化效益、考慮觀點的異同、以及所獲得的加速增益等。研究成果將可提供爲未來相關 PDE's 平行運算求解的一個性能指標。

關鍵字：平行運算；多處理器環境；數值方法；偏微分方程式；雙向波動方程式。

Abstract

In this paper, several numerical schemes, such as Leap Frog, MacCormack, and Runge Kutta of the finite difference schemes, are implemented and applied to solve a set of hyperbolic PDE's under distinct multiprocessor architectures: Sequent Symmetry and Transputer. The PDE to be solved is a two-way wave equation that is used to describe the propagation of waves in a material with discontinuous sound speed. The performance and the efficiency of parallel computation are evaluated based on obtained parallel speedup and serial speedup.

Keywords: Parallel computations; Multiprocessor architecture; PDE; Two-way wave equations.

1. Introduction

In future scientific and engineering problems which involve the solutions of PDE's, computational resources and solution time spent in a traditional serial computer will become extremely large due to the ever growing size of problem scales and the needs of the more accurate resolutions. Scholars with those needs expect to require processing rate far in excess of the limit of the traditional serial computer [1~4]. For the past 30 years, the development of the multiprocessors has motivated researchers' interest in applying parallel processing techniques to the solution of the very large scientific and engineering problems [5,6]. Of course, parallel computation for the PDE's solutions has been one of the fields seeking to take advantage of such achievement. However, it is also true that most of the published researches and results focuses on implementing one or few numerical methods to solve a certain PDE's on a specific parallel architecture may not be commercially available [7~9].

To perform the parallel computations of PDE's in multiprocessor architectures, the *Sequent Symmetry S27* [13] with 6 processors and the *Transputer* with 8 processors were employed and the two-way wave equation of hyperbolic PDE was selected as problem to be solved. Several numerical algorithms were implemented for the solutions of the PDE, including Leap Frog, MacCormack, and Runge Kutta methods. The parallel efficiency for each numerical algorithm can be studied by analyzing the obtained speedup for each case [14].

2. Mathematical Model

In this model, the propagation of waves in a material with discontinuous sound speed will be studied. This problem is actually a model of what the oil companies do when they look for oil. They use the vibration as signal pulse and transmit it underneath. In the surface of the earth, microphones are used to measure the reflected signals. By analyzing the

*This research was supported by the National Science Council of the R.O.C. under grant NSC 84-2212-E-035-007.

different arrivals they try to infer the underground structure.

Consider the two-way wave equation problem that can be expressed in a matrix form:

$$\begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} 0 & \frac{1}{\rho(x)} \\ \rho(x)\cdot c(x)^2 & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix}_x \quad -\pi \le x \le \pi \quad (1)$$

initial conditions:

$$u(x,0) = v(x,0) = \zeta(x)$$

$\zeta(x)$—*Gaussian function*

$$\zeta(x) = e^{\frac{-\sigma^2 x^2}{2}}, \quad \sigma = 10$$

where:

$u(x,t)$—*transmitted pulse*

$v(x,t)$—*reflected pulse*

$\rho(x)$—*density of the material*

$c(x)$—*sound speed*

where u and v are periodic in x

The main reason of choosing above equation as our model for parallel solution becomes clear once the parameter of $\rho(x)$ and $c(x)$ are taken to be discontinuous along $[-\pi, \pi]$. The propagation of waves in a material with discontinuous sound speed can studied subject to the following three different schemes:

(A) $\rho(x)=c(x)=1.0 \quad -\pi \le x \le \pi$

In this case, where $\rho(x)$ and $c(x)$ are assumed to be constant through all x, we should expect the pulse propagates to the left and eventually on the right by periodicity.

(B) $\rho(x) = c(x) = 1.0 \quad |x| < \frac{\pi}{4}$

$$\begin{cases} \rho(x) = 1.0 \\ c(x) = 1.01 \end{cases} \quad |x| \ge \frac{\pi}{4}$$

In this case, when the pulse hits the interface at $x = \left|\frac{\pi}{4}\right|$ there is a transmitted pulse and accompanied by a reflected pulse. However, the reflected is weak because of the small ratio of the two sound speed and this enable us to perform the test of accuracy. Obviously more computations may be necessary for a certain degree of accuracy and, eventually, the solution time may be elongated.

(C) $\rho(x) = c(x) = 1.0 \quad |x| < \frac{\pi}{4}$

$$\begin{cases} \rho(x) = 1.0 \\ c(x) = 3.0 \end{cases} \quad |x| \ge \frac{\pi}{4}$$

In this case, there is a strong reflection when the pulse hits the interface at $x = \left|\frac{\pi}{4}\right|$ due to the larger ratio of the two sound speed, hence, we should know how well can our numerical schemes handle the large discontinuity in $c(x)$. Actually this case is a test of stability, to find the solution in parallel environment while the computations and results become more complicated.

3. Numerical Schemes

Various numerical algorithms, such as *Leap Frog (LF)*, *MacCormack (Mac)*, and *Runge Kutta (RK)* of the finite difference methods are implemented and their performances are evaluated from the view point of parallel efficiency [15,16]. Based on the levels of the solution approximation on the discretized grids, they can also be further categorized into several variants, as was concluded in Table I.

3.1 Leap-Frog Scheme (LF):

LF is a two-stage scheduling and it requires a starting procedure to compute the value at $t=\Delta t$ since only initial condition is available at this particular case. Based on the order taken for central difference in time (t) and space (x), some variations, which are all Neumann stable, in this scheme include LF22, LF24, and LF24 with dissipation. Both LF22 and LF24 are non-dissipation schemes, and in LF24 with dissipation, the approximations in higher order derivatives are enhanced to prevent high frequency oscillation in solution which might be generated by the incorrect solution at $x=\pi$.

3.2 MacCormack Scheme (Mac):

Variations in Mac scheme are made due to the order taken for central difference in time and space, however, they are all single level, Neumann stable, and dissipation scheme. For each Mac22, Mac24, and Mac26 scheme, two different forms are made for forward predictor/backward corrector (FB) and backward predictor/forward corrector (BF).

3.3 Runge Kutta Scheme (RK):

The multistage scheduling in time is the main distinction for RK scheme with LF and Mac schemes, which are both single-stage in time. Hence it takes more memories to store extra data induced by this multistage scheduling in time. Similar to LF scheme, RK is a stable and non-dissipation method and is usually employed with artificial dissipation.

The error criterion used for convergence check can be shown as below:

$$Error = \sqrt{\frac{\sum_{j=1}^{N} \left| u_j^n - u_{exact}(x_j, t^n) \right|^2}{\sum_{j=1}^{N} \left| u_{exact}(x_j, t^n) \right|^2}} \quad (2)$$

where $u_{exact}(x_j, t^n) = \zeta(x_j + t^n)$ is shown as the exact solution of case (A). Of course, the check for convergence in case (B) or case (C) can also be formed simply by replacing it with the corresponding exact solution. It is clear that the error will naturally become lesser with the increased grid points. This may become necessary as a more accurate numerical solution is requested in order to attain a better graphical resolution, especially in case (B) where the reflection is considerably weak.

Table I: Numerical schemes employed for solutions.

Methods	Notation	Variants		Remarks
Leap Frog	LF22	2-2		2^{nd} order in time/2^{nd} order in space
	LF24	2-4		2^{nd} order in time/4^{th} order in space
	LF24a	2-4 with rtificial dissipation		2^{nd} order in time/4^{th} order in space
MacCormack	Mac22	2-2	FB	Forward predictor/Backward corrector
			BF	Backward predictor/Forward corrector
	Mac24	2-4	FB	Forward predictor/Backward corrector
			BF	Backward predictor/Forward corrector
	Mac26	2-6	FB	Forward predictor/Backward corrector
			BF	Backward predictor/Forward corrector
Runge-Kutta	RK	4-stage		Multistage in time

4. Parallel Computations

Among those parameters used to describe or classify the wealth of parallel computers available, the type of processor interconnection is a significant aspect that provides a point of view on the mechanism by which processors exchange information. Generally speaking, there are two major classifications:

1. Share memory architecture,
2. Local memory architecture,

and a variety of hybrid designs lying in between. To fully evaluate the parallel computation efficiency of the numerical algorithms in distinct multiprocessor architectures, the PDE's numerical solution is performed under both environments, including Sequent of shared memory and Transputer of local memory.

4.1 Sequent Symmetry—Shared Memory Machine

This alternative uses a global shared memory that can be accessed by all processors. While each processor executes its own instruction on data in a shared memory, a processor can communicate to another by writing to the global memory, and then having the second processor read the same location in the memory. Apparently it solves the interprocessor communication problem, but introduces the intricacy of simultaneous accessing different locations of the memory by several processors. Under such architecture, algorithm design is simplified since the system behaves as if all the processors were directly connected to each other. The Sequent Symmetry employed is a shared memory machine with 6 processors access the required data through a global memory space. Since the system was implemented in a time-sharing mode, it is difficult to obtain the privilege of accessing all processors without disturbance. In current study, the parallel computations were performed by using 4 processors at most.

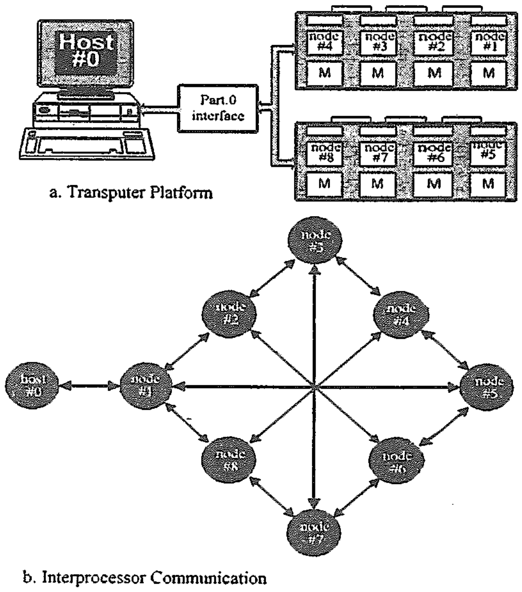

Figure 1: The CSA transputer, (a) interface with PC host and hardware configuration, (b) interprocessor communication links.

4.2 Transputer—Local Memory Machine

There is no common memory in this architecture, but rather each processor has its own local memory. Processors communicate through an interconnection network which usually consists of direct communication links joining certain pairs of processors. Hence, which processors are connected together is an important design issue. Undoubtedly, it would be the best if all processors were directly connected to each other, but this is often not feasible because it would cause either an excessive number of links, which leads to increased cost, or the processors would communicate through a shared bus, which leads to excessive delays as the number of processors is large due to the necessary bus contention.

The CSA transputer used in this study is known as the inexpensive parallel processing tools and is PC formatted which can be installed in PC-style expansion cabinet. The transputer platform of current system includes a PC 486 as a host which connects to the processor board (Part 6) with 8 nodes through an interface card (Part 0). The interprocessor connection is rather flexible and can be performed by way of certain hardwire links and software configuration. However, for current study, the interprocessor links were configured as was shown in Figure 1.

4.3 Performance Measurement

The algorithms of Table I are modified to fit the characteristics of each specific multiprocessor architecture. The parallel speedup and serial speedup, as expressed in eq. (3) and eq. (4), are both examined to relate how well a particular algorithm has been parallelized and to describe the speed advantage of the parallel algorithm in a multiprocessor computer, respectively.

$$\text{parallel speedup} \quad S_p(nproc) = \frac{t_p(1)}{t_p(nproc)} \qquad (3)$$

$$\text{serial speedup} \quad S_s(nproc) = \frac{t_s}{t_p(nproc)} \qquad (4)$$

where t_s is the computation time of serial program with 1 processor; t_p(nproc) is the computation time of parallel program with *nproc* processors for parallel computation.

5. Performance Results

The speedups obtained from both parallel computers will be discussed as both serial and parallel speedup are included with available processor number ranging from 1 to 4 for Sequent, and from 1 to 8 for Transputer while the grid points (NX) is 1000.

5.1 Speedups on Sequent

It is found that the performance of each scheme on Sequent did not change much for three distinct cases, that is, the discontinuity of sound speed *c(x)* at $x = \frac{\pi}{4}$ did not affect the parallel computation efficiency as far as shared memory machine is concerned. For Leap Frog algorithm, the 2-4 scheme with artifical dissipation (LF24a) appeared to have better performance to LF22 and LF24 in both speedup measurement. In MacCormack algorithm, meanwhile, the advantage of using 3 different variations (Mac22, Mac24, Mac 26) was not clear from the view point of speedup performance. However, the Runge Kutta algorithm showed a superior performance in almost every categories compared with every other algorithms. Among three algorithms, it seems that RK has a better performance as far as parallel speedup is concerned. However, it must be known that RK also takes much longer in computation time for the numerical results to meet convergence criterion than LF does, and spent comparable amount of time with MacCormack .

5.2 Speedups on Transputer

It is also noted that the MacCormack variations show the best efficiency while the Leap Frogs pertain the worst speedups among all the algorithms. Note also that the performance of implemented algorithms on three cases are similar except for Leap Frogs which works more efficient on case C. As the Leap Frog algorithms are concerned, LF24a generally performs better than the others. In Contrast to results of Leap Frog schemes, the increased stages of MacCormack scheme does not result in a better speedup although the computation time was greatly elevated for Mac26 and Mac24 about 5 and 4 times than Mac22. As was suggested, Runge Kutta is superior to all variations of Leap Frog while provides less efficiency than every schemes of MacCormack.

5.3 Comparisons

The performance of each algorithm on Sequent is comparable for three different cases, that is, the computation for the discontinuity of sound speed *C(x)* at $x = \frac{\pi}{4}$ did not affect the parallel computation efficiency under shared memory machine. Similar conclusions can also be made for MacCormack's and Runge Kutta on Transputer of local memory machine. However, it was found that variances exist for Leap Frogs on transputer for three distinct cases. The best speedups were obtained on case C while case B seemed to work inefficiently for LF22 and LF24. Interprocessor communication is often one of the main obstacles to increasing performance of parallel algorithms for local memory machines. Among three algorithms, Leap Frogs is expected to spent more interprocessor communication than the others [18].

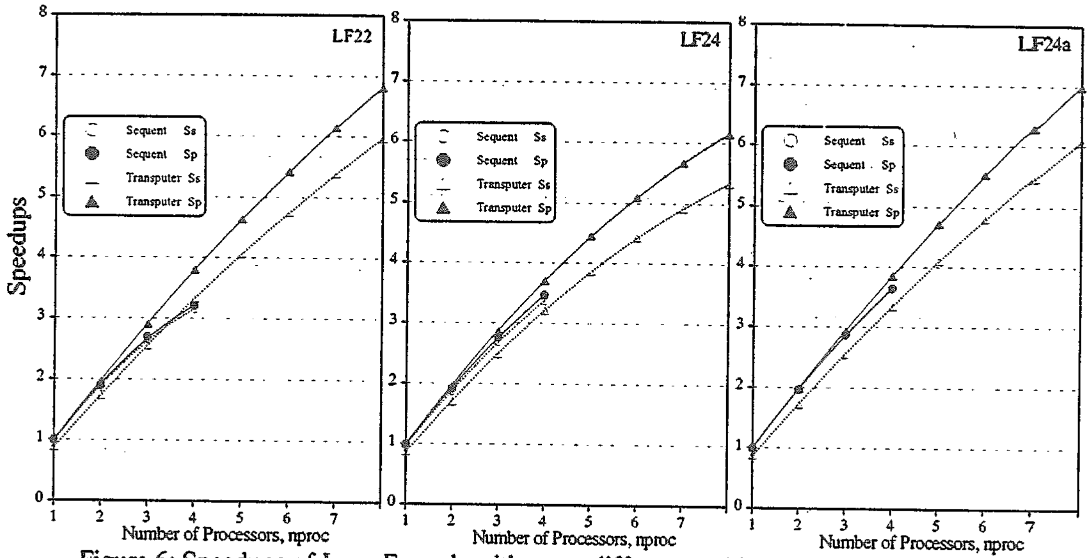

Figure 6: Speedups of Leap Frog algorithms on different multiprocessor environments

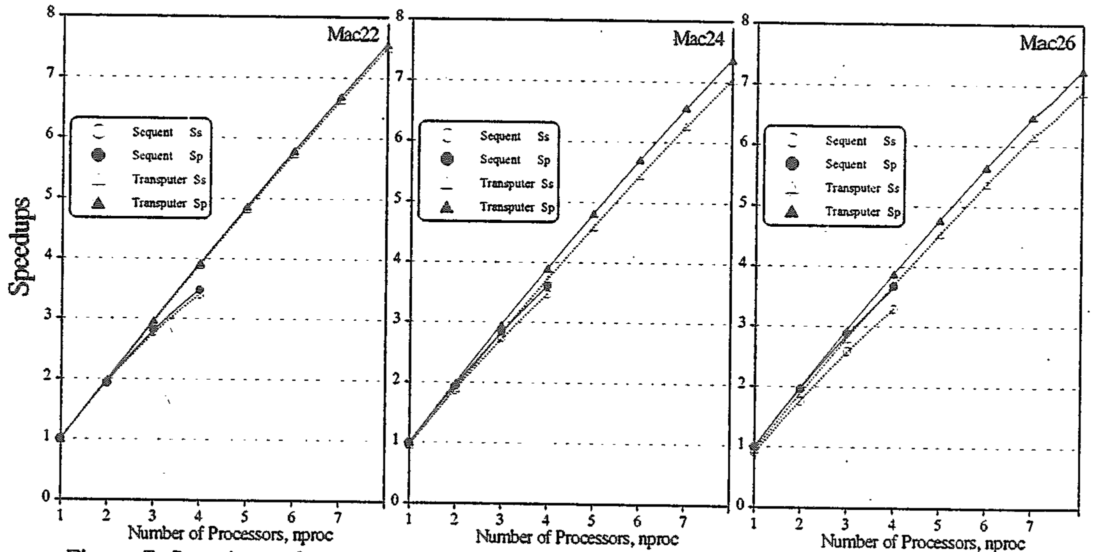

Figure 7: Speedups of MacCormack algorithms on different multiprocessor environments

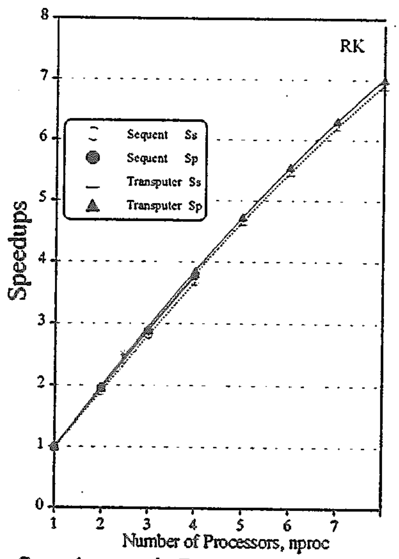

Figure 8: Speedups of Runge Kutta algorithm on different multiprocessor environments

Considering case C with NX=1000, the comparisons for the parallel efficiency of two multiprocessor environments can be concluded in Fig. 2~Fig. 4 for three algorithms and their variances. In the results of Leap Frog schemes, it is found that the gap between parallel speedup and serial speedup on Transputer is comparably large while difference of the two in Sequent is not significant, as was depicted in Fig. 2. Transputer performs more efficient than Sequent does as parallel speedup is concerned although this advantage gradually lessen for LF22 and LF24a. However, it is also interesting to see that, in contrast to the result of parallel speedup, Sequent is slightly superior to Transputer in serial speedup. Observation of Fig. 3 shows that Transputer has better performance than Sequent for both parallel and serial speedup on three MacCormack variations. On the other hand, Transputer also exhibits a consistent performance for processor number ranging from 1 to 8 while Sequent seems to reveal the tendency of degrading with increasing number of processors. It should be noticed that Runge Kutta is not the most time-consuming among all the numerical schemes under studied. Nevertheless, the speedup results depicted in Fig. 4 suggest that RK performs steadily on both shared and local memory machines although it does not present the best performance overall.

6. Conclusions

In this paper, three algorithms and variants from finite difference method were implemented on two distinct multiprocessor computers to solve a two-way wave equation that is used to describe the propagation of waves in a material with discontinuous sound speed. The performance results were examined based on the obtained speedups to determine how well the algorithms were parallelized under certain multiprocessor architecture environment. The results indicated that Leap Frog, MacCormack, and Runge Kutta methods all show fine efficiency in solving the specified PDE's in parallel.

While making a decision of what algorithms to choose for the numerical solution of current PDE's application, some indices other than speedup performance may have been considered, including the accuracy, stability, and difficulties while the selected algorithms is parallelized. Meanwhile, the advantage of using one specific multiprocessor architecture over the other is not clear without making a overall evaluation of all factors, such as computation time, available number of processors, and which index (serial or parallel speedup) to be concerned. However, only LF24a and MacCormacks have assured themselves to the level of stability. On the other hand, Runge-Kutta and LF24 with dissipation are not easy to be parallelized from the parallel computation's viewpoint. Based on the experiments and performance evaluations, it is not unambiguous that the MacCormack is a good choice in general ratings and local memory machine seems to perform more efficient than shared memory machine does while parallel computation becomes the main issue of the case.

References

[1] G. Zorpette, "The Power of Parallelism", *IEEE Spectrum*, Sep. 1992, pp. 28~33.

[2] K. Dowd, *High Performance Computing*, O'Reilly & Associates Inc., 1993.

[3] T.E. Bell, "Beyond Today's Supercomputers", *IEEE Spectrum*, Sep. 1992, pp. 72~75.

[4] J.M. Ortega and R.G. Voigt, *Solution of Partial Differential Equations on Vector and Parallel Computers*, SIAM, 1985.

[5] K. Hwang, *Advanced Computer Architecture*, McGraw-Hill, Inc. 1993.

[6] P. Wiley, "A parallel architecture comes of age at last", *IEEE Spectrum*, June 1987, pp. 46~50.

[7] M.A. Pirozzi, "The fast numerical solution of mildly nonlinear elliptic boundary value problems on multiprocessors", *Parallel Computing*, 19(1993), pp. 1117~1128.

[8] L. Lopez, "Methods based on boundary value techniques for solving parabolic equations on parallel computers", *Parallel Computing*, 19(1993), pp. 979~991.

[9] O.A. McBryan, "The Connection Machine: PDE Solution on 65536 Processors", *Parallel Computing*, 9(1989), pp. 1~24.

[10] R. Duncan, "A Survey of Parallel Computer Architectures", *IEEE Computer*, February 1990, pp. 5~16.

[11] G.R. Desrochers, *Principles of Parallel and Multiprocessing*, McGraw-Hill, Inc. 1988.

[12] J.J. HACK, "On the promise of general-purpose parallel computing", *Parallel Computing*, 10(1989), pp. 261~275.

[13] Sequent Computer Systems, Inc., *Guide to Parallel Programming on Sequent Computer System*, Prentice Hall, 1992.

[14] L.A. Crowl, "How to Measure, Present, and Compare Parallel Performance", *IEEE Parallel and Distributed Technology*, Spring 1994, pp. 9~25.

[15] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth, Inc. 1989.

[16] D. Gottlieb, and S.A. Orszag, *Numerical Methods Analysis of Spectral Methods: Theory and Applications*, SIAM, 1977.

[17] B.H. Shen, *Parallel Computation of the Two-Way Wave Equation in Shared Memory Multiprocessor Architecture*, M.S. Thesis, Dept. of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan, 1995.

[18] S.G. Shu, *Parallel Computation of the Two-Way Wave Equation under Multiprocessor Environment*, M.S. Thesis, Dept. of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan, 1997.